REMARKS

Claims 1-8 are pending in the present application. Claims 1, 2, 5 and 6 are amended above. No new matter is added by the claim amendments. Entry is respectfully requested.

The specification is objected to for failing to state the meaning of the acronym "HPRIF". The specification is amended above to indicate that HPRIF register is defined as a "programmable priority register". No new matter is added by the amendment to the specification. Entry of the amendment and removal of the objection to the specification are respectfully requested.

Claims 1-8 stand rejected under 35 U.S.C. 112, second paragraph for reasons stated in the Office Action at page 2, paragraphs 3-5. Claims 1 and 5 are amended above to state that the pointer information "indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", which clarifies the meaning of the term pointer information. This feature is supported in the specification as filed at page 9, lines 5-17, page 11, line 27 through page 12, line 5 and in the drawings at FIGs. 6-8. Claims 2 and 6 are amended above to state that "the priority information is programmed such that higher weight in the form of increased access to a bus common to the bus masters is given to at least one of the bus masters" which clarifies the meaning of the term weight, and is supported in the specification at page 12, line 18 through page 13, line 4 and in the drawings at FIG. 10. In the example provided in FIG. 10, "master 3" is given higher weight in the round-robin arbitration, since it appears twice in the

register. It is believed that the claim amendments and above remarks overcome the rejections. Entry of the amendments and removal of the rejections are respectfully requested.

Claims 1-8 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Sato, *et al.* (U.S. Patent Number 5,583,999) in view of "C++ Primer" by Stanley Lippman. Claims 1-8 are further rejected under 35 U.S.C. 103(a) as being unpatentable over the applicant's admitted prior art (AAPA) in view of Sato, *et al.* Reconsideration of the rejection and allowance of claims 1-8 are respectfully requested.

In the present invention as claimed in independent claim 1, a programmable fixed priority and round-robin arbiter includes a "rotating unit". The "rotating unit", "when operating in both a fixed priority mode and in a round-robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register, and outputs reordered priority information". The arbiter further includes a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode" and "a single, common grant-reordering unit used for operation in both the fixed priority mode and in the round-robin mode".

In the present invention as claimed in independent claim 5, in a bus control method, an "arbiter" rotates "priority information related to bus masters stored in a register to give the

highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register, and outputting the reordered priority information". The method further includes the arbiter "reordering requested priorities of the bus masters corresponding to the reordered priority information in both the fixed priority mode and the round-robin mode of operation".

Sato, *et al.* discloses, in embodiment 2, a bus arbiter which selects one of a linear method of arbitration and a round-robin method of arbitration by switching between the activation and deactivation of priority registers 17-1, 17-2, under control of a control register 15a (see Sato, *et al.*, FIG. 4 and column 8, lines 30-35). If the priority registers 17-1, 17-2 are deactivated, priority determining units 11-1, 11-2 prioritize bus use according to the linear method of arbitration, and if the priority registers 17-1, 17-2 are activated, priority determining units 11-1, 11-2 prioritize bus use according to the round-robin method of arbitration. Sato, *et al.* further discloses, in embodiment 3, a bus arbiter that arbitrates bus use according to a hybrid of both the linear method and the round-robin method by activating and deactivating the priority registers 17-1, 17-2. In this embodiment, priority of master 2-3 is set higher than the remaining bus masters 2-1, 2-2 and 2-4 through 2-8, which, in turn, are evenly distributed according to the round-robin method of arbitration.

It is stated in the Office Action at page 4, line 8, that Sato, *et al.* does not disclose the practice of a pointer. In follows then, that Sato, *et al.* fails to teach or suggest an arbiter that includes a "rotating unit", "when operating in both a fixed priority mode and in a round-robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", as claimed in claim 1. Sato, *et al.* further fails to teach or suggest a method that includes an "arbiter rotating priority information related to bus masters stored in a register to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register", as claimed in claim 5.

It is stated in the Office Action, at page 4, lines 1-3, that Sato, *et al.* discloses that the priority determining units 11-1, 11-2 prioritize the bus masters, and that such function is equivalent to the claimed request-reordering unit. When both the linear method and the round-robin method are used in Sato, *et al.*, the priority determining unit 11-1 is used for the linear method and 11-2 is used for the round-robin method. Therefore, Sato, *et al.* fails to teach or suggest a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode", as claimed in claim 1.

"C++ Primer" by Stanley Lippman discloses that a pointer variable holds values that are the addresses of objects in memory, and that through a pointer, an object can be referenced indirectly. While the Applicant does not dispute that pointers are useful programming tools, "C++ Primer" by Stanley Lippman, like Sato, *et al.*, in no way teaches or suggests an arbiter that includes a "rotating unit", "when operating in both a fixed priority mode and in a round-robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", as claimed in claim 1. "C++ Primer" by Stanley Lippman, like Sato, *et al.*, further fails to teach or suggest a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode", as claimed in claim 1. "C++ Primer" by Stanley Lippman, like Sato, *et al.*, further fails to teach or suggest a method that includes an "arbiter rotating priority information related to bus masters stored in a register to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register", as claimed in claim 5.

Neither Sato, *et al.* nor "C++ Primer" by Stanley Lippman teaches or suggests an arbiter that includes a "rotating unit", "when operating in both a fixed priority mode and in a round-

robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", as claimed in claim 1. In addition, neither Sato, *et al.* nor "C++ Primer" by Stanley Lippman teach or suggest a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode", as claimed in claim 1. Further, neither Sato, *et al.* nor "C++ Primer" by Stanley Lippman teaches or suggests a method that includes an "arbiter rotating priority information related to bus masters stored in a register to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register", as claimed in claim 5. Accordingly, it is submitted that the combination of Sato, *et al.* and "C++ Primer" by Stanley Lippman fails to teach or suggest the invention as claimed in claims 1 and 5. Reconsideration of the rejection of, and allowance of, claims 1 and 5 under 35 U.S.C. 103(a) as being unpatentable over Sato, *et al.* and "C++ Primer" by Stanley Lippman are respectfully requested. With regard to the dependent claims 3-4, 6-8, it follows that these claims should inherit the allowability of the independent claims from which they depend.

With regard to the rejection of claims 1-8 as being unpatentable over the AAPA in view of Sato, *et al.*, the AAPA discloses that, when operating in a fixed priority mode, the priorities are arranged based on information stored in an HPRIF register and that, when operating in a round-robin mode, the priorities are arranged based on pointer information. Therefore, only the round-robin mode of arbitration utilizes pointer information in the AAPA. Further, the AAPA discloses the use of separate request-reordering and request-rotating logic units 110, 160 for the fixed priority mode and the round-robin mode, which leads to the limitations described in the specification as filed at page 4, line 20 through page 5, line 12.

The AAPA fails to teach or suggest an arbiter that includes a "rotating unit", "when operating in both a fixed priority mode and in a round-robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", as claimed in claim 1. Further, the AAPA fails to teach or suggest a method that includes an "arbiter rotating priority information related to bus masters stored in a register to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register", as claimed in claim 5. Instead, in the AAPA, pointer information is

used only in the round-robin mode and not in the fixed priority mode. Further, AAPA fails to teach or suggest a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode", as claimed in claim 1. Instead, in the AAPA, separate request-reordering and request-rotating logic units 110 and 160 are used for the fixed priority mode and the round-robin mode, respectively.

Neither the AAPA nor Sato, *et al.*, as discussed above, teaches or suggests an arbiter that includes a "rotating unit", "when operating in both a fixed priority mode and in a round-robin mode, rotates information related to bus masters stored in a register in a direction of rotation to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority, the pointer information defining the amount of rotation to be initiated by the rotating unit so that reordered priority information is available in the register", as claimed in claim 1. Further, neither the AAPA nor Sato, *et al.* teach or suggest a "single, common request-reordering unit used for operation in both the fixed priority mode and in the round-robin mode", as claimed in claim 1. Further, neither the AAPA nor Sato, *et al.* teaches or suggests a method that includes an "arbiter rotating priority information related to bus masters stored in a register to give the highest priority to a bus master in response to pointer information which indicates the bus master currently having the highest priority in both the fixed priority mode and the round-robin mode, the pointer information defining the amount of rotation to be initiated so that reordered priority information is available in the register", as claimed in claim 5. Accordingly, it is submitted that the combination of the AAPA and Sato, *et al.* fails to teach or

suggest the invention as claimed in claims 1 and 5. Reconsideration of the rejection of, and allowance of, claims 1 and 5 under 35 U.S.C. 103(a) as being unpatentable over the AAPA and Sato, *et al.* are respectfully requested. With regard to the dependent claims 3-4, 6-8, it follows that these claims should inherit the allowability of the independent claims from which they depend.

Closing Remarks

It is submitted that all claims are in condition for allowance, and such allowance is respectfully requested. If prosecution of the application can be expedited by a telephone conference, the Examiner is invited to call the undersigned at the number given below.

Respectfully submitted,

Date: December 1, 2005

Mills & Onello, LLP
Eleven Beacon Street, Suite 605
Boston, MA 02108
Telephone: (617) 994-4900, Ext. 4902
Facsimile: (617) 742-7774

J:\SAM\0490\Amend A\amendmenta.wpd

Anthony P. Onello, Jr.
Registration Number 38,572
Attorney for Applicant